

Markov Decision Process based Model for Performance Analysis an Intrusion Detection System in IoT Networks

Gauri Kalnoor and Gowrishankar S

BMS College of Engineering, Bangalore, India

<https://doi.org/10.26636/jit.2021.151221>

Abstract—In this paper, a new reinforcement learning intrusion detection system is developed for IoT networks incorporated with WSNs. A research is carried out and the proposed model RL-IDS plot is shown, where the detection rate is improved. The outcome shows a decrease in false alarm rates and is compared with the current methodologies. Computational analysis is performed, and then the results are compared with the current methodologies, i.e. distributed denial of service (DDoS) attack. The performance of the network is estimated based on security and other metrics.

Keywords—DDoS, intrusion detection, IoT, machine learning, Markov decision process (MDP), Q-learning, NSL-KDD, reinforcement-learning.

1. Introduction

The technology of the Internet of Things (IoT) is relatively new, it connects the Internet to the low hardware resources devices and then susceptible to the various malicious attack, i.e. denial of service (DOS) [1], [2]. The network-based IoT is considered to be one the fastest evolving areas, having 50 billion gadgets connected among them [3], and then vulnerable to security abuse. For example, Mirai is one of the unusual types of a botnet which triggers a large-scale attack like distributed denial-of-service (DDoS) and thus strikes by mistreating some of the IoT devices [4], and even infects the CCTV IP cameras [5].

The safety of IoT is constantly improved [6]. Many frameworks and methods are developed to mitigate most network attacks. The logs with recorded abuse historical data are observed, based on methods using machine learning which can reach a large network – up to millions in a day.

The intrusion detection system (IDS) is an essential component in the security of the network to protect the target network which comprises of irregular actions and threats during interruption of network traffic. Thus, there is a separation of normal activity and anomalous activity in the network. A comprehensive IDS group can be obtained in two classes. Misuse-based IDS is the interrupt that notices

the known strategies. The limit of the primary technique to anticipate new and obscure assaults is restricted. The signature-based IDS is dependent on the irregularity identification and works by making a profile of ordinary conduct of the network, then later recognizing it as any anomalous conduct [3].

In the proposed work, an artificial intelligence (AI) based algorithm has been proposed for developing an IDS for detection of malicious attacks and also monitors the data streams generated from IoT and WSNs [6].

It is an enhanced method of Markov decision process with Q-Network algorithm which gives an optimal best solution in terms of performance of IoT networks. Thus, it is an important and challenging issue to be considered, and decision modeling is applied to obtain the optimal solution. The main contributions of this article are summarized below:

- the RL-based IDS is proposed by exploiting the extended Markov decision process (MDP) algorithm,
- the RL calculation is consolidated on IDS (RL-IDS) with the end goal that the survey for cases like a basic foundation is obtained by unique digital-based hazards for IoT and WSN continuously,
- a Q-network is applied with the end goal that the assessment of Q-work is recognized by conveying IDS into RL. A few tests are performed for the assessment of the execution of the proposed model in the environment considered.

The remaining sections of this paper are presented as follows. Section 2 describes the related work. Section 3 introduces the method for security and reinforcement learning. In Section 4, the system model is formulated and RL-IDS methodology is described. In Section 5, performance is investigated and results are presented. In Section 5, the evaluation carried out for the proposed RL-based IDS scheme is explained and then compared in Section 6 with supervised machine learning schemes. Lastly, the work concludes with the experiments and analysis in Section 7.

2. Related Work

In recent works, many authors have applied standard techniques of machine learning (ML), such as principal component analysis (PCA) and linear discriminant analysis (LDA), as these classification-based algorithms can detect normal records with high precision and identify the abnormal records such that the performance of an IDS can be managed [7]–[11]. In [12], the authors have proposed deep feature embedding to reduce the size or magnitude of data from the network based on IoT in a real-time application by considering the “edge of deep learning”. Likewise, in [13] the preprepared worldview is applied such that the identification and quickness are helped with traditional ML-based calculations.

In [14], the authors have observed that the IoT technology makes possible to connect different smart objects, through the Internet. The authors have formulated a novel QoS management schemes based on power control algorithm. The unexplored R-learning algorithm is used as a doctive paradigm by the authors where the system agents teach other agents to adjust the power levels, thus reducing the complexity in computation and increasing speed in the learning process.

In [15] the optimization has been incorporated into an MDP which can minimize the evaluation metric as long-term average delay. The continuity of state and action space due to the high dimensionality is considered by the author where deep reinforcement learning based dynamic resource management (DDRM) algorithm is proposed. This enables the joint optimization with computing resource and transmission power. The authors have compared the simulated results with conventional URM, RRM and A3C algorithms mainly which reduces the delay in task effectively.

Also, taking as an illustration of the idea-based IDS, Q-learning of reinforcement learning (RL) has been investigated thoroughly by examining and protecting the sensor network that utilizes the dynamic methodology and ideal activities based on the arrangement of states in the respective IoT environment [16]. There are numerous papers on scientific classification, position, and the ML current advancements in data security, i.e. [17], [18]. Structured [19] ML techniques have been applied to location interruption for network information. The exemplary ML models applied to IDS were: support vector machine (SVM), multi-layer perceptron (MLP), k-nearest neighbors (KNN), decision trees (DT), naive Bayes (NB), and random forest.

3. Security in IoT

To meet the ideal security necessities, a complete perspective on network security is required. The accompanying key security properties ought to be viewed when building up a convincing IoT security methodology.

- **confidentiality** – it is a crucial security standard for IoT structures. IoT devices can store and move sensi-

tive information that shouldn't be wrongly found by individuals [21],

- **authentication** – the verification of both communication parties must be completed before performing other procedures,
- **integrity** – the IoT applications need the legitimate constituents to be uniquely altered where the information is moved through the remote correspondence,
- **availability** – the authorized users should be consistently able to access the IoT network,
- **authorization** – this includes granting privileges to clients for an IoT structure [22],

3.1. Reinforcement Learning-based IDS

Beginning by characterizing the idea of RL, and other augmentation of ML dependent on Markov decision process (MDP), first a reward function R is defined providing state s to IDS. It is characterized with five IDS concepts as below.

System state space. The arrangement of states gained by the IDS is $S = s_0 - \text{ordinary}, s_1 - \text{identification}, s_2 - \text{no detection}$, where s_0 demonstrates the typical traffic record in the WSN record, s_1 implies the location of IDS assaults on traffic, and s_2 demonstrates that IDS can't recognize assaults.

Action space. A set of possible actions that the IDS can perform, can be expressed by:

$$A = \{a_0, a_1, a_2, a_3, \dots, a_m\} , \tag{1}$$

where a_k indicates the type of IDS reaction in the k -th attack class and $k = 0, 1, 2, \dots, m, p$, for example, according to Table 1. The shares are sorted according to their risk level: $a_0 < a_1 < a_2 < a_3 < \dots < a_m$.

Table 1
Known attacks and their risk level

Risk	Attack instances
Low	Gues-passwd, Warezclient, FTP-write
Medium	Satan, Portsweep, Nmap
High	DNS-poisoning, Cross-site-scripting (XSS), ARP-spoofing
Critical	ICMP flood, Land, Smurf, Ping of death, Apache 2

Reward function. The rewarded function is negative when the IDS makes the best move to secure the framework regardless of whether the scheme against the activity is too costly, and positive when the IDS chooses the right activity.

The estimation of the reward is:

$$r_t(s_t, a_t) = \left\{ \begin{array}{l} R_p \text{ for } s_t=0 \text{ and } a_t=a_0 \\ 1-\mu_j(a_t)R_p \text{ for } s_t=s_0 \text{ and } a_t \in \{a_1, \dots, a_m\} \\ R_p \text{ for } s_t=s_1 \text{ and } a_t=a_k \\ 1-\lambda_j(a_t)R_p \text{ for } s_t=s_1 \text{ and } a_t \in \{a_0, \dots, a_{k-1}\} \\ R_n \text{ for } s_t=s_1 \text{ and } a_t \in \{a_{k+1}, \dots, a_m\} \\ R_p \text{ for } s_t=s_2 \text{ and } a_t=a_m \\ 1-\theta_j(a_t)R_p \text{ for } s_t=s_2 \text{ and } a_t \neq a_0 \end{array} \right\}, \quad (2)$$

where $0 < \mu_j(a_j) < 1$, $0 < \lambda_j(a_t) < 1$ and $0 < \theta_j(a_t) < 1$. The r_t refer to the reward s_t is the state of the sensor node, a_t is the action of the sensor at t time.

The reward in each time t is:

$$r_t(S_t = s, a_t = a) = \sum_{s' \in S} P\left(\frac{s}{s'}, a\right) r_t(s', a) \quad (3)$$

State transition probability. The transition probability matrix at time t for $a \in A$ is:

$$\mathbf{P}_a = \begin{bmatrix} \beta_{1,1}^a & \beta_{1,2}^a & \beta_{1,3}^a \\ \beta_{2,1}^a & \beta_{2,2}^a & \beta_{2,3}^a \\ \beta_{3,1}^a & \beta_{3,2}^a & \beta_{3,3}^a \end{bmatrix}. \quad (4)$$

Given by β^a :

$$i, j = p\left(s_t + \frac{1}{s_t}\right) = p\left(\frac{s_t}{s_j}, a\right) \text{ for } i, j = 1, 2, 3.$$

$$\sum_{j=1}^3 \beta_{i,j}^a = 1, i = 1, 2, 3 \text{ and } a \in A. \quad (5)$$

Discount factor. $0 < \gamma < 1$. The IDS arbitrarily choose a_t , and the environment samples the reward $r_t(s_t, a_t)$ according to the state of arrival s . The agent then receives an incentive in the following state s_{t+1} . Besides, π is a specific policy from s_t to s_{t+1} specifying a_t retrieved in each state s_t . Then, the strategy is updated to generate sample paths (s_0, a_0, r_0) , (s_1, a_1, r_1) , $(s_2, a_2, r_2) \dots$. Let us define $\pi = (\pi_1, \pi_2, \dots)$ as the best policy vector. The goal of the data stream is to get π_t , which represents the best pattern based on system status. Therefore, the expected maximum sum of IDS rewards at t , is given by:

$$\pi^* = \arg \max_{a \in A} [r_t(s_t, a_t) + \sum_{s' \in S} P_t(s'|s, a) V_{t-1-t}(s')]. \quad (6)$$

The optimal value function V_{i+1} defines the IDS which can be chosen as the best state. It can be found out from each phase:

$$V_{i+1}(s) = \arg \max_{a \in A} [r_{t-1-t}(s_t, a_t) + \sum_{s' \in S} P_{t-1-t}(s'|s, a) V_i(s')]. \quad (7)$$

Next, the timestamp size is determined, using the concept of Q-learning. In every state, the best action a is chosen and the algorithm Q-learning applied, so that the updates can be performed. The optimal policy π^* is calculated according to the best action. If there are no optimal actions found, then the learning samples $0 < \alpha < 1$ are applied.

$$Q(s_t - a_t) = Q(s_t, a_t) + \alpha [r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)]. \quad (8)$$

The pair (s, a) is updated to determine the step having the best reward. In each iteration, the prediction of IDS has state value function V_{i+1} and then a Q-table is constructed by using Q-learning, where the lines signify the columns and states s representing the actions a . In each state s_t , the reward r_t is observed corresponding to an action a_t realized by the agent. The action at the next state (s_{t+1}) is also observed in [21], and the approximate value of Q is updated to satisfy the Bellman equation:

$$Q(s_{t+1} - a_{t+1}) = (1 - \alpha)Q(s_t, a_t) + \alpha [r_t + \gamma \max_{a' \in A} Q(s', a')]. \quad (9)$$

4. Proposed Model

The random forests (RF) algorithm is used to classify a large amount of data. Several algorithms like decision trees and merging trees are used during classification to train the sample data available. The final output during classification chooses the most selected class [7].

In this section, the details of the deployment of the Q-learning network-based model are provided aiming to monitor and predict the cyber-attacks in critical infrastructures of sensed big data streams. The discussion is encompassed in the following aspects:

- the attack risks and their different degree,
- the pre-processing details engaged to clean data and filter,
- the strategy of the interaction of IDS model by the agent to secure the attacks,
- the Q-function estimation and its results by considering the best decision.

The architecture of the proposed system is shown in Fig. 1, which presents the sensor data of WSN and the RL-IDS mechanism requested to make a decision.

At pre-processing stage, the network traffic is registered for every type of attack and then invalid and redundant records are removed. Next, the transformation of the record is done based on the type of attack [9]. At the first step, data aggregation obtained by the sensor [20] is performed so that the data volume is reduced.

Next, the Q-network (QN) is applied by using the Q-function for estimation of best action to the attack. It improves the prediction and the estimation of action values effectively among the state's set by applying the non-linear function: $Q(s_t, a_t; \theta) \approx \mathcal{Q}(s_{t+1}, a_{t+1})$.

The θ represents neuron weights to be changed by the end of each iterative step i . The implementation of Q-network is further improved by:

- utilizing a step forward for the present state s to get predictive Q values,
- applying the replay (like historical IDS for the interactive process) into data let $Ht =$

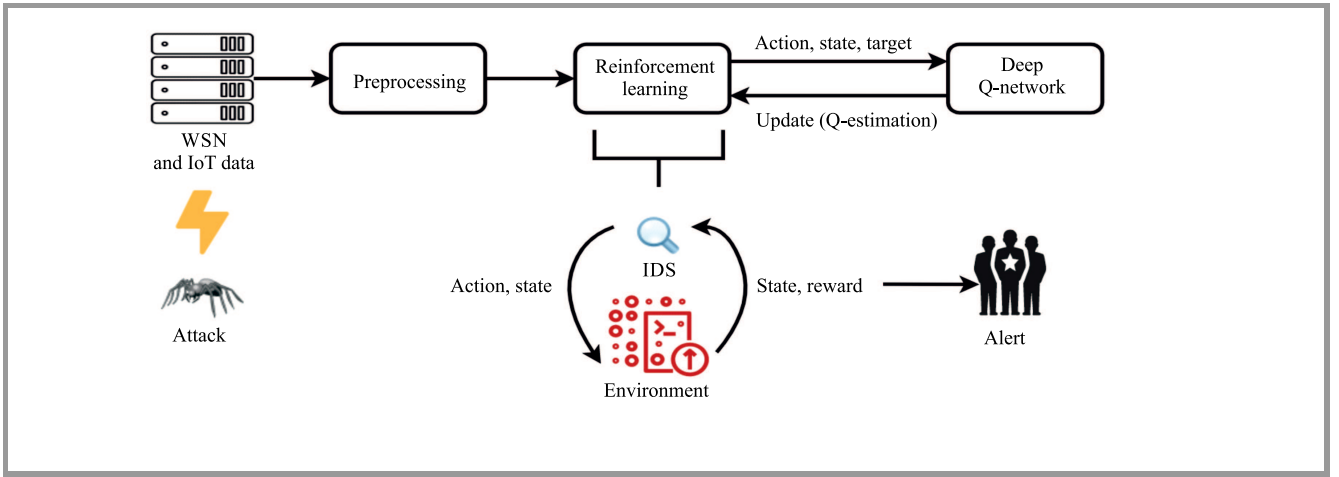


Fig. 1. Proposed method of improvement the IoT and WSN based RL-IDS.

$\{h(1), h(2), \dots, h(t)\}$ within an over-time t as $f_t = (s_t, a_t, r_t, s_{t+1})$,

- updating the Q-network based on the data from training (r, s, a, s) over the target Q-value with optimization of the loss-function during an iterative step noted as:

$$L_i(\theta_i) = E\{[x_i - Q(s, a, \theta_i)]^2\}, \quad (10)$$

$$x_i = r_t + \gamma \arg \max_{a'} Q(s', a', \theta_{i-1}), \quad (11)$$

- applying back-propagation with loss function's gradient, the weights are updated corresponding to the θ parameters as:

$$\nabla_{\theta_i} L_i(\theta_i) = E\{[x_i - Q(s, a, \theta_i)] \nabla_{\theta_i} Q(s, a, \theta_i)\}. \quad (12)$$

4.1. Model Description

In the proposed scheme, the problem for QoS control is tackled based on the approach of R-learning algorithm. The main aim of every QoS scheduler is maximization the amount of data transmitted with low power consumption. For this fundamental trade-off, the function U is defined to analyze the ratio of throughput to power. Thus, the function for QoS scheduler at i -th position U_i is:

$$U_i(B_j^i, B_{-i}) = \frac{TS_i(B)}{B_j^i}, \text{ s.t., } B_j^i \in B_i, B = \prod_{i \in N} B_i | B_i \in [B_1^i, B_m^i], \quad (13)$$

where B_{-i} is the transmit power vector without B_i , and $TS_i(B)$ is the throughput scheduler.

In wireless communication, the signal to interference noise ratio (SINR) in the given effective range γ_i is measured while computing the throughput at i -th scheduler TS_i and can be expressed using:

$$TS_i(B) = W \cdot \log_2 \left(1 + \frac{\gamma_i(A)}{\Omega} \right), \quad (14)$$

where W is referred to as bandwidth of the channel assigned in through IoT network, Ω ($\Omega \geq 1$) is the gap between capacity and the uncoded M-ary quadrature amplitude modulation (M-QAM).

Algorithm 1: The IoT-WSN-based RL-IDS used for training and testing

Data: sensor data dataset Y

Input: Initialize action, state, environment, parameter θ , targeted Q-network
Initialize reply-memory H space

Output: return vector $Q(s_t, a_t, \theta)$

```

while  $|\widehat{Q}_{i+1} - \widehat{Q}_i| < \sigma$  do
  for  $X = 1, 2, 3, \dots, N$  do
     $s_0$  = starting state
    for  $t = 0, 2, 3, \dots, T - 1$  do
      Select an action (random)  $a_t$  with
      a random-probability  $p$  based on  $\in$ 
      strategy as:
       $a_t = \arg \max_a Q(s, a_k, \theta)$ 
      - Apply  $a_t$  and the reward observed by the
      IDS- $r_t$  and the next state observe chosen
      reward  $r_t$  and store the tuple
       $(s_t, a_t, r_t, s_{t+1})$  in  $H$ 
      - Arbitrary batch selection with this
      selected feature  $(s_t, a_t, r_t, s_{t+1})$  from  $H$ 
      if  $s_{t+1}$  terminal state then
        |  $\mu l = rl$ 
      end
      else
        |  $\mu l = rl + \delta \arg \max_{a'} Q(s', a', \theta)$ 
      end
      Gradient calculation of the loss function
      based on Eq. (11)
    end
  end
end
    
```

Table 2
Dataset used for evaluation

Category	Port	Attack	Tools	Size [bytes]
Information collect		Scanning of service OS fingerprinting	Nmap, hping3, xprobe2 Nmap	1.4 MB 358 KB
Denial of service	UDP, TCP HTTP	Distributed DoS	hping3 golden-eye hping3	19.5 MB 18.8 MB 19.7 KB
	TCP, HTTP UDP	DoS	hping3 hping3 golden-eye	11.2 MB 21.7 MB 29.7 KB
Information theft		Key-logging data theft	Metasploit	1369
			Metasploit	118

The environment was made by consolidating traffic and Table 2 shows the used datasets and software tools.

5. Evaluation Criteria

The validation of proposed algorithm is researched by two measures:

- **Accuracy** – this metric is measured as the degree of closeness between the actual and the predicted value,
- **Precision** – this is a metric that describes the accuracy level obtained from the mentioned information and the outcomes anticipated by the executed model. Consequently, accuracy is the proportion of true positive forecasts contrasted with general aftereffects of positive expectation.

Table 3 shows the boundaries or limits used for CNN and MLP algorithms.

Table 3
Parameters of algorithms used for testing

Algorithm	Batch size	Function (activation)	Optimizer	Epochs
Convolution neural network (CNN)	32,64,128	Softmax, ReLu	Adam	10, 30, 50
Multilayer perceptron (MLP)	32,64,128	Softmax, ReLu	Adam	10, 30, 50
Markov decision process (MDP)	32,64,128	Softmax, ReLu	Adam	10, 30, 50

A major drawback of any IoT sensor network is that these devices work in remote networks and have to be sustained on their battery life. Hence the average energy consumed by the device plays a vital role which depends on its performance as shown in Fig. 2, the node shows that the MDP algorithm provides a less amount of energy consumption when compared with CNN and MLP algorithm. MDP provides significant results as false detection is reduced even when the number of nodes is increased as shown in Fig. 3. As the number of nodes increases the false detection is getting reduced as compared with MLP and CNN.

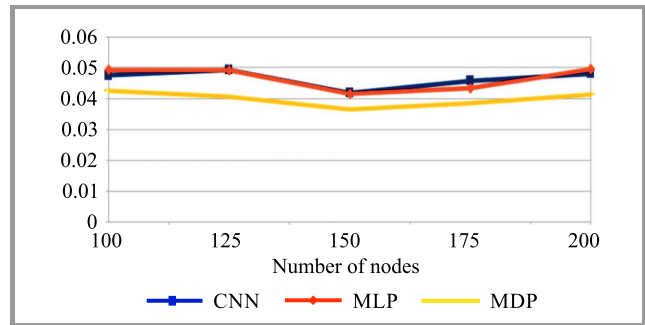


Fig. 2. Average energy consumption by number of nodes.

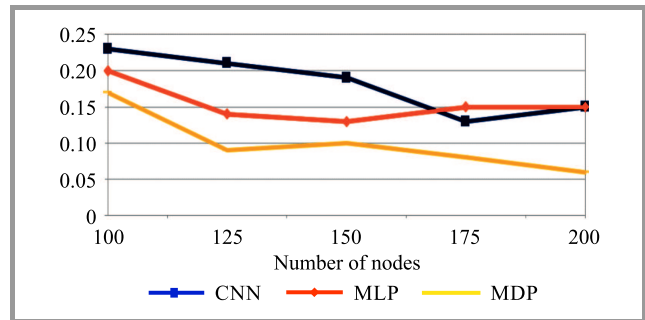


Fig. 3. False alarm rate.

The system of IoT mainly in a wireless system depends on the success rate of message delivery even when the number of nodes are increased and have a successful delivery rate which is provided in Fig. 4. In this plot all algorithm with the proposed algorithm, the throughput is given and can be observed that the MDP performance is good for throughput when nodes are more.

A comparison figure of the detection rate of IoT systems is shown in Fig. 5 which depicts that the detection rate at the receiver node in MDP is better when compared with CNN and MLP.

Figure 6 presents normalized overhead for several nodes in the IoT network when compared with all other algorithms with the reinforced algorithm MDP, it provides better performance for normalized overhead when compared with MLP and CNN. Parameters from Table 4 were used in this plot.

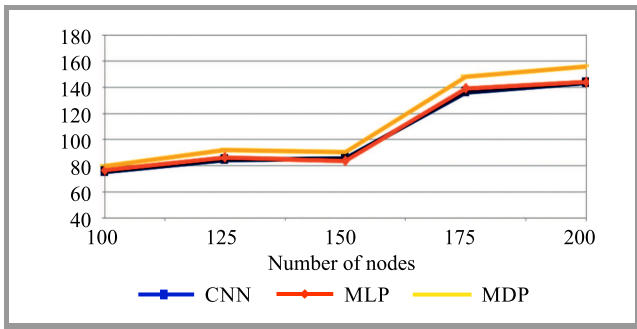


Fig. 4. Throughput rate of change.

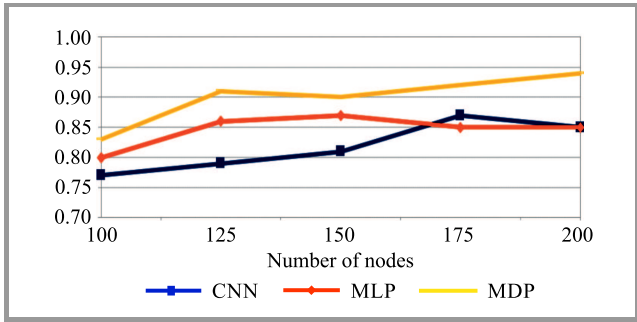


Fig. 5. Detection rate.

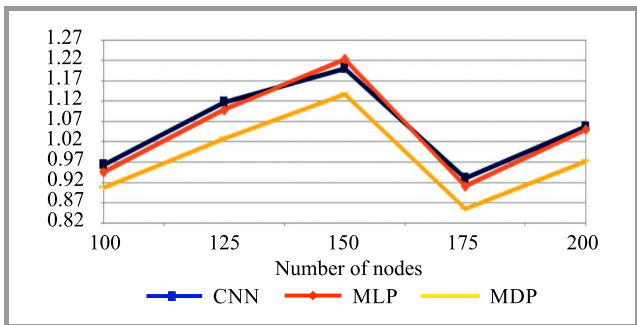


Fig. 6. Normalized overhead.

Table 4 Evaluation metrics (detection rate of attacks)

Algorithm	Metrics				
	DDoS attack	DoS attack	Reconnaissance	Normal (AUC)	Theft (AUC)
MDP	0.99	0.99	0.97	0.99	0.95
CNN	0.98	0.97	0.98	0.98	0.99
MLP	0.55	0.49	0.96	0.97	0.97

Table 5 represents the classification and comparison results based on the feature selection and the AUC precision metrics.

In Table 6, the mean accuracy is expanded as the number of study ages for the MLP classifier. For CNN, there was a decrease as the quantity (in terms of numbers) of epochs increased from 10 to 50.

Table 7 shows the same accuracy evaluation for size of 64. For this situation, the accuracy (batch size 64) diminished

Table 5 Comparison analysis

Algorithm	AUC	Precision	Sensitivity
MDP	0.99	99.80%	98.55%
CNN	0.92	96.75%	97.00%
MLP	0.89	95.05%	93.02%

Table 6 The accuracy evaluation for batch size 32

Algorithm	Epoch	Mean Accuracy	Elapsed time
MDP	10	93.22%	60 min 12 s
CNN	10	91.75%	58 min 39 s
MLP	10	54.07%	39 min 09 s
MDP	30	91.03%	165 min 25 s
CNN	30	89.72%	158 min 30 s
MLP	30	63.95%	124 min 33 s
MDP	50	90.00%	230 min 21 s
CNN	50	89.30%	229 min 22 s
MLP	50	63.00%	186 min 47 s

with the expansion epochs for the classifier (MLP). Data decreasing a bit while the number of epochs is increased from 10 to 50 in CNN.

Table 7 Accuracy for batch size 64

Algorithm	Epoch	Mean accuracy	Elapsed time
MDP	10	92.00%	18 min 40 s
CNN	10	91.15%	20 min 57 s
MLP	10	76.92%	26 min 56 s
MDP	30	92.30%	62 min 17 s
CNN	30	91.02%	64 min 18 s
MLP	30	54.04%	64 min 19 s
MDP	50	92.30%	114 min 60 s
CNN	50	90.64%	112 min 55 s
MLP	50	53.89%	102 min 20 s

Table 8 shows the outcome for block size of 128. The normal exactness seems to increment along with the expanding number of the experiment of epochs for MLP-based classifier. For the CNN, a slight diminishing was observed as the number of epochs rises from 10 to 30. In all cases the larger batch size the shorter application lifetime.

6. Conclusion

In the proposed work, the reinforcement learning in a network is examined. The valuation of the RL-IDS model is incorporated and compared with different ML and DL algorithms such as CNN and LP. The RL calculation gave the best outcome and precision and AUC leads in multiclass

Table 8
Mean accuracy for batch size 128

Algorithm	Epoch	Mean accuracy	Elapsed time
MDP	10	92.50%	12 min 12 s
CNN	10	90.87%	11 min 33 s
MLP	10	54.10%	10 min 16 s
MDP	30	93.00%	40 min 50 s
CNN	30	90.76%	45 min 44 s
MLP	30	54.43%	27 min 58 s
MDP	50	92.03%	55 min 27 s
CNN	50	91.27%	54 min 27 s
MLP	50	79.01%	46 min 18 s

characterization. With epoch increase a slight reduction in precision is observed, while in the 128-batch preliminaries, there was an increase in accuracy. A double change in MLP could make the estimation cycle 1.4 to 2.6 s faster, while CNN could make the figuring cycle 1.8 to 2.4 s shorter. Later on, the models with various calculations are likely created and different calculations for AI or profound learning are joined. Moreover, this calculation ought to be actualized in NIDS so it very well may be utilized progressively to alleviate attacks.


References

- [1] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks", in *Proc. IEEE 9th Annual Comput. and Commun. Workshop and Conf. (CCWC)*, Las Vegas, NV, USA, 2019, pp. 452–457 (DOI: 10.1109/CCWC.2019.8666588).
- [2] X. Yuan, C. Li, and X. Li, "DeepDefense: identifying DDoS attack via deep learning", in *Proc. of the 2017 IEEE Int. Conf. on Smart Comput. (SMARTCOMP)*, Hong Kong, China, 2017, pp. 1–8 (DOI: 10.1109/SMARTCOMP.2017.7946998).
- [3] D. Evans, "The Internet of Things: how the next evolution of the Internet is changing everything", *Cisco Internet Business Solutions Group (IBSG)*, 2011 [Online]. Available: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- [4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets", *Computer*, vol. 50, no. 7, 2017, pp. 80–84 (DOI: 10.1109/MC.2017.201).
- [5] P. Radanliev *et al.*, "Future developments in cyber risk assessment for the Internet of Things", *Computers in Industry*, vol. 102, pp. 14–22, 2018 (DOI: 10.1016/j.compind.2018.08.002).
- [6] E. Bertino and N. Islam, "Botnets and Internet of Things security", *Computer*, vol. 50, no. 2, pp. 76–79, 2017 (DOI: 10.1109/MC.2017.62).
- [7] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, and M. Guizani, "A Survey of machine and deep learning methods for Internet of Things (IoT) security", *IEEE Commun. Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020 (DOI: 10.1109/COMST.2020.2988293).
- [8] A. Okwori, "Intrusion detection in Internet of Things (IoT)", *Int. J. of Advanced Res. in Computer Sci.*, vol. 9, pp. 504–509, 2018 (DOI: 10.26483/ijarcs.v9i1.5429).
- [9] Y. Meidan, "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis", in *Proc. of the Symp. on Applied Comput. – SAC '17*, Marrakech, Morocco, 2017, pp. 506–509 (DOI: 10.1145/3019612.3019878).
- [10] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A Supervised intrusion detection system for smart home IoT devices", *IEEE Internet of Things J.*, vol. 6, no. 5, 2019, pp. 9042–9053 (DOI: 10.1109/JIOT.2019.2926365).
- [11] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for Internet of (battlefield) things devices using deep eigenspace learning", *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, 2019, pp. 88–95 (DOI: 10.1109/TSUSC.2018.2809665).
- [12] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of Things: A comprehensive investigation", *Comput. Netw.*, vol. 160, pp. 165–191, 2019 (DOI: 10.1016/j.comnet.2019.05.014).
- [13] R. Nicolescu *et al.*, "Mapping the values of IoT", *J. Inf. Technol.*, vol. 33, pp. 345–360, 2019 (DOI: 10.1057/s41265-018-0054-1).
- [14] S. Sheng *et al.*, "Deep reinforcement learning-based task scheduling in IoT edge computing", *Sensors (Basel)*, vol. 21, no. 1666, 2021 (DOI: 10.3390/s21051666).
- [15] Y. Chen *et al.*, "Deep reinforcement learning based dynamic resource management for mobile edge computing in industrial Internet of Things", *IEEE Transac. on Industrial Informat.*, vol. 17, no. 7, pp. 4925–4934, 2021 (DOI: 10.1109/TII.2020.3028963).
- [16] M. Elrawy, A. Awad, and H. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey", *J. Cloud Comput.*, vol. 7, no. 21, 2018 (DOI: 10.1186/s13677-018-0123-6).
- [17] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2013 (DOI: 10.1109/SURV.2013.052213.00046).
- [18] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: current solutions and future challenges", *arXiv [Online]*. Available: <https://arxiv.org/pdf/1904.05735.pdf>
- [19] K. A. P. da Costa, J. P. Papa, C. de Oliveira-Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: a survey on machine learning-based intrusion detection approaches", *Computer Networks*, vol. 151, pp. 147–157, 2019 (DOI: 10.1016/j.comnet.2019.01.023).
- [20] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection", in *2018 Wireless Telecommun. Symp. (WTS)*, Phoenix, AZ, USA, 2018, pp. 1–5 (DOI: 10.1109/WTS.2018.8363930).
- [21] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things", *IEEE Access*, vol. 7, pp. 42450–42471, 2019 (DOI: 10.1109/ACCESS.2019.2907965).
- [22] M. Abomhara and G. M. Koen, "Cyber security and the Internet of Things: vulnerabilities, threats, intruders and attacks", *J. of Cyber Secur. and Mobil.*, vol. 4, no. 1, pp. 65–88, 2015 (DOI: 10.13052/jcsm2245-1439.4).



Gauri Kalnoor received her B.E. and M.Tech. from the department of Computer Science and Engineering, Visvesaraya Technological University, Belgavi in 2008 and 2010, respectively. She has worked in Central University of Karnataka as an Assistant Professor and in Wipro Technologies as a Project Engineer. She is a re-

search scholar in B.M.S.C.E. Research Centre and her research area is Internet of Things. She is interested in coding and analysis of machine learning techniques.

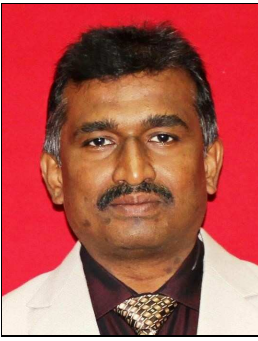
 <https://orcid.org/0000-0001-9970-4697>

E-mail: kalnoorgauri@gmail.com

B.M.S.C.E

Basavangudi

Bangalore, India



Gowrishankar S is a senior Professor at Computer Science & Engineering department at BMS College of Engineering, Bangalore. He served as a Head of the department of CS&E and IS&E of BMSCE. He is actively associated with the Research Collaborative Sabbatical program with University of Alabama, Huntsville (UAH), USA

and he is a visiting professor for UAH. Having an Academic and Research experience of 20 years, he authored more than 80 research publications in reputed international journals and conferences. His research interests include performance evaluation, wireless network and deep learning.

 <https://orcid.org/0000-0002-8119-8711>

E-mail: Gowrishankar.cse@bmsce.ac.in

B.M.S.C.E

Basavangudi

Bangalore, India